

MPP 媒体处理软件

FAQ

文档版本 V1.0

发布日期 2024-12-21

前言

概述

本文为使用 MPP 媒体处理软件开发的程序员而写，目的是为您在开发过程中遇到的问题提供解决办法和帮助。



说明

未经特殊说明，以 XM7606V13 指代 XM76 系列所有芯片。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本	操作系统

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
 注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

寄存器访问类型约定

类型	说明	类型	说明
RO	只读，不可写。	RW	可读可写。
RC	读清零。	WC	可读，写 1 清零，写 0 保持不变。

寄存器复位值约定

在寄存器定义表格中：

- 如果某一个比特的复位值 “Reset”（即 “Reset” 行）为 “？”，表示复位值不确定。
- 如果某一个或者多个比特的复位值 “Reset” 为 “？”，则整个寄存器的复位值 “Total Reset Value” 为 “-”，表示复位值不确定。

数值单位约定

数据容量、频率、数据速率等的表达方式说明如下。

类别	符号	对应的数值
数据容量（如 RAM 容量）	1K	1024
	1M	1,048,576
	1G	1,073,741,824
频率、数据速率等	1k	1000
	1M	1,000,000
	1G	1,000,000,000

地址、数据的表达方式说明如下。

符号	举例	说明
0x	0xFE04、0x18	用 16 进制表示的数据值、地址值。
0b	0b000、0b00000000	表示 2 进制的数据值以及 2 进制序列（寄存器描述中除外）。
X	00X、1XX	在数据的表达方式中，X 表示 0 或 1。 例如：00X 表示 000 或 001； 1XX 表示 100、101、110 或 111。

其他约定

本文档中所用的频率均遵守 SDH 规范。简称和对应的标称频率如下。

频率简称	对应的标称频率
19M	19.44MHz
38M	38.88MHz
77M	77.76MHz
622M	622.08MHz

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2024-12-21	V1.0	DI 版本发布。

目 录

前 言	i
1 系统控制	1
1.1 日志信息	1
1.1.1 如何查看 MPP 的日志信息	1
1.2 内存使用	2
1.2.1 OS 保留内存和线程栈大小调整	2
1.3 性能相关	3
1.3.1 CPU 性能 Top 统计波动大问题	3
1.3.2 绑定中断到不同 CPU 的注意事项	3
1.4 修改内核选项后重编 KO 流程	3
2 视频输入	4
2.1 VI 时序配置	4
2.1.1 DC 配置	4
2.1.1.1 VI DEV 配置	4
2.1.1.2 VI PIPE 配置	4
2.1.2 MIPI 配置	5
2.1.2.1 VI DEV 配置	5
2.1.2.2 VI PIPE 配置	5
2.1.3 BT.601 配置	6
2.1.3.1 VI DEV 配置	6
2.1.3.2 VI PIPE 配置	6
2.1.4 BT.656 配置	7
2.1.4.1 VI DEV 配置	7

2.1.4.2 VI PIPE 配置	7
2.1.5 BT.1120 配置.....	8
2.1.5.1 VI DEV 配置	8
2.1.5.2 VI PIPE 配置	8
2.2 调试命令	8
3 视频编码.....	10
3.1 VENC	10
3.1.1 JPEG 量化表配置注意事项	10

1 系统控制

1.1 日志信息

1.1.1 如何查看 MPP 的日志信息

【现象】

需要查看日志和调整 log 日志的等级。

【分析】

Log 日志记录 SDK 运行时错误的原因、大致位置以及一些系统运行状态等信息。因此可通过查看 log 日志，辅助错误定位。

目前日志分为 8 个等级（0~7），默认设置为等级 3。等级设置的越高，表示记录到日志中的信息量就越多，当等级为 7 时，系统的整个运行状态实时的被记录到日志中，此时的信息量非常庞大，会大大降低系统的整体性能。因此，通常情况下，推荐设置为等级 3，因为此时只有发生错误的情况下，才会将信息记录到日志中，辅助定位绝大多数的错误。

【解决】

获取日志记录或修改日志等级时用到的命令如下：

- 查看各模块的日志等级，可以使用命令 `cat /proc/umap/log`，此命令会列出所有模块日志等级。
- 修改某个模块的日志等级，可使用命令 `echo 'venc=4' > /proc/umap/log`，其中 `venc` 是模块名，与 `cat` 命令列出的模块名一致即可。
- 修改所有模块的日志等级，可以使用命令 `echo 'all=4' > /proc/umap/log`。

- 获取日志记录，可以使用命令 **cat /dev/log**，此命令将打印出所有的日志信息；如果日志已读空，命令会阻塞并等待新的日志信息，可以使用 Ctl+C 退出。如果不想阻塞等待日志信息，可以使用命令 **echo wait=0 > /proc/umap/log** 取消阻塞等待。也可以使用 **open**、**read** 等系统调用来操作/dev/log 这个设备节点。

1.2 内存使用

1.2.1 OS 保留内存和线程栈大小调整

【现象】Linux 系统跑业务程序出现 oom-killer。

【分析】可能的原因有

- OS 内存不足
- 系统保留内存太小

【解决】

- 增大 OS 内存
- 增大系统保留内存，可以在/etc/profile 中加入以下命令将系统保留内存设置为 4M（大小可调整）

```
echo 2 > /proc/sys/kernel/randomize_va_space
```

```
echo 4096 > /proc/sys/vm/min_free_kbytes
```

【现象】

系统跑业务程序出现创建线程失败打印 pthread_create: Resource temporarily unavailable 的错误信息。

【分析】可能的原因有

- OS 内存不足
- 线程栈空间太大

【解决】

- 增大 OS 内存
- 调整线程最大栈空间大小，调整方式有 2 种：

- Linux 系统可以使用 `ulimit -s` 命令修改线程栈大小，例如将线程栈大小设置成 1M：`ulimit -s 1024`，可以在 `/etc/profile` 中加入此命令则可以开机就设置栈空间大小。
- 使用 `pthread_attr_setstacksize` 在程序中改变线程栈大小。

1.3 性能相关

1.3.1 CPU 性能 Top 统计波动大问题

【现象】使用 top 进行 cpu 占用率统计不是很准确，可能出现波动，特别是在小业务场景，top 统计的 cpu 占用率波动会很大。

【分析】版本 linux kernel 默认使用 HZ 为 100，也即为 10ms 调度统计，统计时间粒度较粗，导致统计精度不够，如此波动会比较大。

【解决】如果期望比较准确的 cpu 占用率统计值，可以修改 kernel HZ 为 1000，如此可以提高统计精度。

1.3.2 绑定中断到不同 CPU 的注意事项

针对中断绑定 CPU 的操作有如下建议：

- 绑定 CPU 的操作要在业务运行之前进行，不要在业务运行过程中动态切换绑定；
- 同一个模块的多个核要绑在同一个 CPU 上；
- 把中断比较多的模块识别出来绑定到其它 CPU 上，比如网络的中断若比较多，可以把它跟媒体业务分开。

1.4 修改内核选项后重编 KO 流程

【现象】客户有修改内核选项的需求，有些选项修改后 KO 需要重新编译。

【分析】无

【解决】

- 客户修改内核选项后回到 SDK 根目录，重新整编 SDK 即可。
- 由于内核/驱动/文件系统三者强关联，建议用户修改内核配置选项后重新整编 SDK，不要只编译内核。

2 视频输入

2.1 VI 时序配置

VI 支持 DC、MIPI、BT.601、BT.656、BT.1120 等时序输入，不同的时序输入需要使用对应的 DEV 以及对应的时序配置，才能保证视频通路能够正常解析不同时序的数据。

2.1.1 DC 配置

2.1.1.1 VI DEV 配置

- 接口类型：XMEDIA_INTF_TYPE_DC
- 接口与 VI 引脚连接关系：正确连接，无需配置；错误连接：
 - enable = XMEDIA_TRUE。
 - 连接顺序错误的引脚：vi_pin[n] = m，vi 引脚 n 接收 DC/BT 输出的第 m bit 数据。
 - 连接顺序正确的引脚：配置默认值，vi_pin[n] = n。

```
xmedia_vi_dc_attr vi_dc_dev_attr =
{
    XMEDIA_INTF_DC_DATA_TYPE_RAW_10BIT,
    {XMEDIA_FALSE}
};
```

2.1.1.2 VI PIPE 配置

- PIPE 的 pipe_bypass_mode 设置为 XMEDIA_VI_PIPE_BYPASS_NONE。
- PIPE 的像素格式设置为 XMEDIA_VIDEO_PIXEL_FMT_RAW。
- PIPE 的位宽根据实际输入设置。

```
xmedia_vi_pipe_config vi_dc_pipe_attr =
{
    XMEDIA_VI_PIPE_BYPASS_NONE,
    XMEDIA_FALSE,
```

```

XMEDIA_FALSE,
{1920, 1080},
XMEDIA_VIDEO_PIXEL_FMT_RAW,
XMEDIA_VIDEO_COMPRESS_MODE_NONE,
XMEDIA_VIDEO_DATA_WIDTH_12,
{-1, -1}
};

```

2.1.2 MIPI 配置

2.1.2.1 VI DEV 配置

- 接口类型: XMEDIA_INTF_TYPE_MIPI_CSI。
- MIPI 属性输入数据类型 input_data_type 根据输入的数据类型设置, 若输入为 RAW 则设置为对应位宽的 RAW 例如 XMEDIA_INTF_MIPI_CSI_DATA_TYPE_RAW_10BIT, 若输入 YUV422 则设置为 XMEDIA_INTF_MIPI_CSI_DATA_TYPE_YUV422_8BIT, 输入为 YUV420 则设置为 XMEDIA_INTF_MIPI_CSI_DATA_TYPE_YUV420_8BIT。

```

xmedia_vi_mipi_attr vi_mipi_dev_attr =
{
    {3840, 2160},
    XMEDIA_INTF_MIPI_CSI_DATA_TYPE_RAW_10BIT,
    XMEDIA_VI_MIPI_LANE_DIVIDE_MODE_4_LANE,
    XMEDIA_VIDEO_WDR_FMT_NONE
};

```

2.1.2.2 VI PIPE 配置

- 若 MIPI 输入的数据类型为 YUV, PIPE 的 pipe_bypass_mode 可设置为 XMEDIA_VI_PIPE_BYPASS_BE, 为 RAW 设置为 XMEDIA_VI_PIPE_BYPASS_NONE。
- PIPE 的像素格式根据 MIPI 实际输入设置。
- PIPE 的位宽根据根据 MIPI 实际输入设置。

```

xmedia_vi_pipe_config vi_mipi_pipe_attr =
{
    XMEDIA_VI_PIPE_BYPASS_NONE,
    XMEDIA_FALSE,
    XMEDIA_FALSE,
    {3840, 2160},
    XMEDIA_VIDEO_PIXEL_FMT_RAW,
};

```

```

XMEDIA_VIDEO_COMPRESS_MODE_NONE,
XMEDIA_VIDEO_DATA_WIDTH_12,
{-1, -1}
};

```

2.1.3 BT.601 配置

2.1.3.1 VI DEV 配置

- 接口类型：XMEDIA_INTF_TYPE_BT601。
- 时序参数：时序参数配置请参考视频输入章节 xmedia_vi_sync_cfg 的说明。
- YUV 顺序：YUV 顺序根据实际输入时序确定：
 XMEDIA_INTF_BT601_BT656_DATA_SEQ_UYVY
 XMEDIA_INTF_BT601_BT656_DATA_SEQ_VYUY
 XMEDIA_INTF_BT601_BT656_DATA_SEQ_YUYV
 XMEDIA_INTF_BT601_BT656_DATA_SEQ_YVYU。

```

xmedia_vi_bt601_attr vi_bt601_dev_attr =
{
    XMEDIA_INTF_BT601_BT656_DATA_SEQ_YVYU,
    {
        XMEDIA_VI_SYNC_PULSE_POSITIVE,
        XMEDIA_VI_SYNC_PULSE_POSITIVE,
        {0, 0}
    },
    {XMEDIA_FALSE}
};

```

2.1.3.2 VI PIPE 配置

- PIPE 的 pipe_bypass_mode 设置为 XMEDIA_VI_PIPE_BYPASS_BE。
- PIPE 的像素格式设置为
 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_422。
- PIPE 的位宽设置为 XMEDIA_VIDEO_DATA_WIDTH_8。

```

xmedia_vi_pipe_config vi_bt601_pipe_attr =
{
    XMEDIA_VI_PIPE_BYPASS_BE,
    XMEDIA_FALSE,
    XMEDIA_FALSE,
    {720, 480},
    XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_422,
    XMEDIA_VIDEO_COMPRESS_MODE_NONE,
};

```

```

XMEDIA_VIDEO_DATA_WIDTH_8,
{-1, -1}
};

```

2.1.4 BT.656 配置

2.1.4.1 VI DEV 配置

- 接口类型: XMEDIA_INTF_TYPE_BT656。
- 时钟采样模式: 根据实际输入确定: 单沿采样/双沿采样。
- YUV 顺序: YUV 顺序根据实际输入时序确定:
XMEDIA_INTF_BT601_BT656_DATA_SEQ_UYVY
XMEDIA_INTF_BT601_BT656_DATA_SEQ_VYUY
XMEDIA_INTF_BT601_BT656_DATA_SEQ_YUYV
XMEDIA_INTF_BT601_BT656_DATA_SEQ_YVYU。

```

xmedia_vi_bt656_attr vi_bt656_dev_attr =
{
    XMEDIA_VI_CLK_EDGE_SINGLE,
    XMEDIA_INTF_BT601_BT656_DATA_SEQ_UYVY,
    XMEDIA_VI_MULTIPLEX_MODE_1,
    {XMEDIA_FALSE}
};

```

2.1.4.2 VI PIPE 配置

- PIPE 的 pipe_bypass_mode 设置为 XMEDIA_VI_PIPE_BYPASS_BE。
- PIPE 的像素格式设置为
XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_422。
- PIPE 的位宽设置为 XMEDIA_VIDEO_DATA_WIDTH_8。

```

xmedia_vi_pipe_config vi_bt656_pipe_attr =
{
    XMEDIA_VI_PIPE_BYPASS_BE,
    XMEDIA_FALSE,
    XMEDIA_FALSE,
    {720, 480},
    XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_422,
    XMEDIA_VIDEO_COMPRESS_MODE_NONE,
    XMEDIA_VIDEO_DATA_WIDTH_8,
    {-1, -1}
};

```

2.1.5 BT.1120 配置

2.1.5.1 VI DEV 配置

- 接口类型: XMEDIA_INTF_TYPE_BT1120。
- YUV 顺序: YUV 顺序根据实际输入时序确定:
XMEDIA_INTF_BT1120_DATA_SEQ_VUVU
XMEDIA_INTF_BT1120_DATA_SEQ_UVUV。

```
xmedia_vi_bt1120_attr vi_bt1120_dev_attr =  
{  
    XMEDIA_INTF_BT1120_DATA_SEQ_VUVU,  
    XMEDIA_VI_MULTIPLEX_MODE_1,  
    {XMEDIA_FALSE}  
};
```

2.1.5.2 VI PIPE 配置

- PIPE 的 pipe_bypass_mode 设置为 XMEDIA_VI_PIPE_BYPASS_BE。
- PIPE 的像素格式设置为
XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_422。
- PIPE 的位宽设置为 XMEDIA_VIDEO_DATA_WIDTH_8。

```
xmedia_vi_pipe_config vi_bt1120_pipe_attr =  
{  
    XMEDIA_VI_PIPE_BYPASS_BE,  
    XMEDIA_FALSE,  
    XMEDIA_FALSE,  
    {1920, 1080},  
    XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_422,  
    XMEDIA_VIDEO_COMPRESS_MODE_NONE,  
    XMEDIA_VIDEO_DATA_WIDTH_8,  
    {-1, -1}  
};
```

2.2 调试命令

可通过以下命令查看、设置 mipi_rx、vicap、viprocs 工作时钟频点

查看 mipi_rx 工作时钟频点:

cat /sys/devices/mipi_rx/devfreq/mipi_rx/cur_freq

查看 vicap 工作时钟频点:

cat /sys/devices/vi/devfreq/vi/cur_freq

查看 viproc 工作时钟频点:

cat /sys/devices/viproc/devfreq/viproc/cur_freq

设置 mipi_rx 工作时钟频点:

echo 475000 > /sys/devices/mipi_rx/devfreq/mipi_rx/userspace/set_freq

设置 vicap 工作时钟频点:

echo 475000 > /sys/devices/vi/devfreq/vi/userspace/set_freq

设置 viproc 工作时钟频点:

echo 475000 > /sys/devices/viproc/devfreq/viproc/userspace/set_freq

3 视频编码

3.1 VENC

3.1.1 JPEG 量化表配置注意事项

JPEG 编码，如果配置的 qfactor 过低，会导致编码出来的 JPEG 图片出现偏色等现象，原因是色度的量化步长过大。用户可以通过调用 `xmedia_venc_set_jpeg_param` 接口修改色度的量化表，限制色度的量化步长，避免偏色等现象。具体的 qfactor 与量化表的关系请见 RFC2435 标准。修改色度的量化表有可能会导导致 JPEG 图片容量变大，用户需要权衡图像质量和 JPEG 图像容量。